Target to Source Coordinate-wise Adaptation of Pre-trained Models

Luxin Zhang[1,2]✉, Pascal Germain[2,3], Yacine Kessaci[1], and Christophe Biernacki[2]

[1] Worldline, France
{luxin.zhang, yacine.kessaci}@worldline.com
[2] MODAL Team, Inria, Lille, France
{luxin.zhang, germain.pascal, christophe.biernacki}@inria.fr
[3] Université Laval, Québec, Canada
pascal.germain@ift.ulaval.ca

A Supplementary Material

Amazon Review Adaptation We consider in total 12 target-source combination pairs, for each category, we train an SVM classifier with linear kernel, the hyperparameter C whose range is from 10^{-5} to 10^4 in a logarithmic scale is fine-tuned using a cross-validation. We use the LinearSVM classifier proposed by the python package Scikit-learn [3]. The classifier is trained using the annotated training dataset and evaluated on the test dataset. Since we have exactly balanced classes, the performance evaluation is based on the precision metric. Because the optimal transport does not have a training phase, the adaptations of our proposed methods and other competitors are performed between the training dataset of the source domain and the test dataset of the target ones.

Kaggle Fraud Detection Adaptation We train the source model in a 4 folds cross-validation way, the source dataset is shuffled and separated into 4 parts. In each experimentation, we use 3 parts to train the source model and use the left 1 part to estimate the hyperparameter. The target dataset is always the same for different source models. For each pair of target-source, we use two pre-trained source models, one gradient boosting decision trees (GBDT) model which is trained using the LightGBM package [1] and one neural network model which is coded in Pytorch [2] to show that our adaptation method is model-independent. When training neural networks, the embedding layer is used to transform categorical features to numerical ones. Because the fraud detection dataset is extremely imbalanced, classical performance metrics like precision could not reveal the real performance of models. Here, we use the precision-recall AUC (PR-AUC).

Regarding the feature selection, we select respectively 1% and 10% of target labels and perform the feature selection using the method proposed in Section 3.5. For GBDT source models, we repeat the feature selection process on 10 different sample sets. For neural networks source model, we repeat the process

30 times. We compare the proposed feature selection adaptation method with other classical domain adaptation methods and the ones with only numerical features adaptation and only categorical features adaptation.

We compare the proposed feature selection adaptation method with other classical domain adaptation methods and the ones with only numerical features adaptation and only categorical features adaptation in the following subsection.

Real Fraud Detection Adaptation The experiment design of fraud detection adaptation is a very realistic scenario. We consider the Belgian data as the source domain and the German data as the target domain. Same as the Kaggle fraud detection adaptation, we train a GBDT model and a neural network model. When training neural networks, the embedding layer is used to transform categorical features to numerical ones. The performances of the adaptation are also evaluated using PR-AUC.

The split of training, validation and test dataset is shown as Fig. 1. For each month, the first two weeks of data are used for supervised learning and adaptation. LightGBM models need fine-tuning for most of the hyperparameters, thus a validation dataset is mandatory. The source classification model is carefully tuned on the left out validation dataset and tested on the test dataset. The entire dataset is not shuffled to prevent the label leak issue. Target domain data are also split using the same setting, regarding the adaptation, we look for a transformation $\mathcal{G}$ between the test set of the target domain and the train set of the source domain.

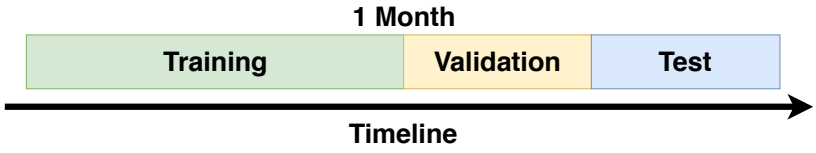

Fig. 1. The split of the fraud detection dataset.

In the production environment, to select the best features for adaptation, we leverage the few labeled transactions in the target domain. To simulate this situation, we use the following step to build the labeled feature selection dataset.

1. We select respectively 1% and 10% of cardholders from ones having at least one fraud transaction in the target domain and let the true labeled information of all their transactions to be known.
2. We select several cardholders from the ones that are not chosen in the first step by keeping the proportion of fraudsters/cardholders to be the same as the production. We annotate all their transactions as genuine. It is possible that among these transactions there are fraud ones. However, it is a very realistic assumption, in the production environment, without the feedbacks of clients, all transactions are considered genuine.

Then we perform the feature selection and performance comparison as the Kaggle dataset. The design of the feature selection dataset is different from the Kaggle fraud detection dataset, because there is not a clear user ID in the Kaggle dataset.

Table 1. Amazon review adaptation results using SVM on 400 dimensional dataset

		No Retrain		
Domains	**Source**	**CORAL**	**OT**	**1D OT**
$B \leftarrow D$	.7334	**.7406**	.6818	.7398
$B \leftarrow E$	.7368	.7463	.7099	**.7479**
$B \leftarrow K$	.7354	.7749	.7155	**.7767**
$D \leftarrow B$	.7034	**.7316**	.6743	.7274
$D \leftarrow E$	**.7715**	.7618	.7113	.7607
$D \leftarrow K$	.7737	.7877	.7121	**.7900**
$E \leftarrow B$	.6945	.6987	.6656	**.7099**
$E \leftarrow D$	**.7203**	.7150	.6787	.7172
$E \leftarrow K$	.7816	.7934	.7426	**.7971**
$K \leftarrow B$	.6898	.7032	.6680	**.7075**
$K \leftarrow D$	.7049	**.7222**	.6795	.7194
$K \leftarrow E$	.7738	.7815	.7354	**.7838**

References

1. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: NeurIPS (2017)
2. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
3. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. JMLR (2011)

Table 2. Amazon review adaptation results using neural networks on 400 dimensional dataset

		Retrain		**No Retrain**		
Domains	**Source**	**DANN**	**DAN**	**CORAL**	**OT**	**1D OT**
$B \leftarrow D$	.7216	.6966±.0210	.6776±.0248	**.7314**	.6820	.7269
$B \leftarrow E$	.6901	.7565±.0086	.6396±.0793	.7496	.7185	**.7635**
$B \leftarrow K$	.6684	.7726±.0067	.6361±.0728	.7660	.7305	**.7757**
$D \leftarrow B$	.7104	.6893±.0081	.7009±.0212	**.7142**	.6709	.7137
$D \leftarrow E$	.7539	.7538±.0114	.7402±.0449	.7546	.7171	**.7632**
$D \leftarrow K$	.7486	.7576±.0037	.7360±.0649	.7702	.7172	**.7776**
$E \leftarrow B$	.6521	.6730±.0184	**.6951±.0022**	.6884	.6647	.6918
$E \leftarrow D$	.6921	.6881±.0174	.7151±.0024	.7021	.6737	**.7163**
$E \leftarrow K$	**.7862**	.7796±.0068	.7842±.0011	.7783	.7372	.7848
$K \leftarrow B$	.6873	.6879±.0023	.6823±.0020	**.6907**	.6734	.6882
$K \leftarrow D$	.7030	.7046±.0022	.7020±.0028	.6993	.6712	**.7147**
$K \leftarrow E$	.7482	**.7795±.0005**	.7524±.0039	.7766	.7275	.7741

Table 3. PR-AUC scores for domain adaptation models and non adaptive models (annotated by a † mark) on Kaggle Fraud Detection dataset (complement of Table **??**).

GBDT model						
Retrain	**Method**	$\%n$	**Model3**	d	**Model4**	d
YES	Train on target†	1	.5569±.0431	-	.5509±.0386	-
	Train on target†	10	**.7369±.0216**	-	**.7310±.0205**	-
NO	Source model†	0	.6819	-	.6775	-
	CORAL NUM	0	.5351	112	.5314	112
	1D OT NUM	0	.6558	112	.6510	112
	1D OT CATE	0	.7034	8	.7026	8
	1D OT	0	.6756	120	.6754	120
	1D OT (ws)	1	.6953±.0115	15±3	.6958±.0063	15±7
	1D OT (ws)	10	.7097±.0042	37±3	.7091±.0022	38±5

Neural Network model						
Retrain	**Method**	$\%n$	**Model3**	d	**Model4**	d
YES	Train on target†	1	.3780±.0589	-	.3721±.0677	-
	Train on target†	10	.5955±.0559	-	.6019±.0423	-
	DAN	0	.6397	120	.6526	120
	DANN	0	.6249	120	.6352	120
NO	Source model†	0	.6548	-	.5848	-
	CORAL	0	.6358	120	.5480	120
	1D OT NUM	0	.6468	23	.5691	23
	1D OT CATE	0	.6687	8	.6260	8
	1D OT	0	.6661	120	.6217	120
	1D OT (ws)	1	.6680±.0090	15±5	.6257±.0151	15±7
	1D OT (ws)	10	**.6884±.0034**	42±9	**.6564±.0052**	42±7

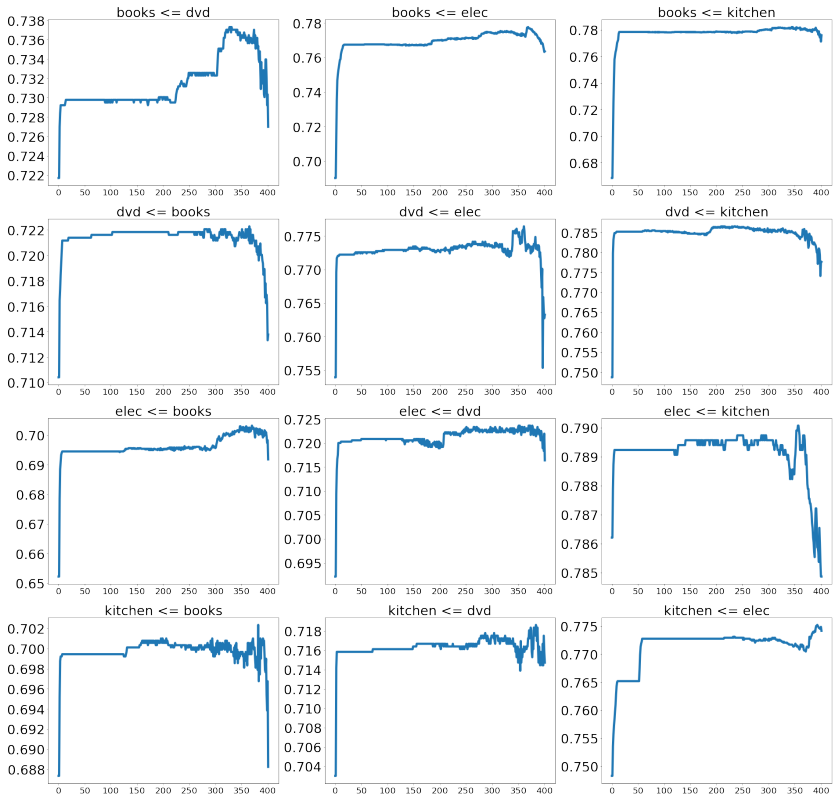

Fig. 2. The evolution of performances according to selected features on Amazon Reviews dataset. Instead of adapting all features, adapting solely well-selected features have better prediction performance. The graph is obtained using a greedy search algorithm with all target label.

Fig. 3. T-sne view of adapted points from Electronics domain to Books domain of Amazon Dataset, showing a 2D projection of the high-dimensional features. The figures show the original features, and those adapted thanks to our proposed method. The graph on the left shows that the adapted Electronics domain overlaps with the books domain and the right graph shows every target-adapted target mapping pairs.

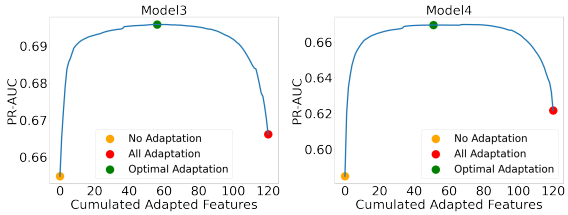

Fig. 4. Feature selection greedy algorithm with neural networks source model on Kaggle fraud detection task (idealized scenario involving all target labels) — (complement of Figure 1).

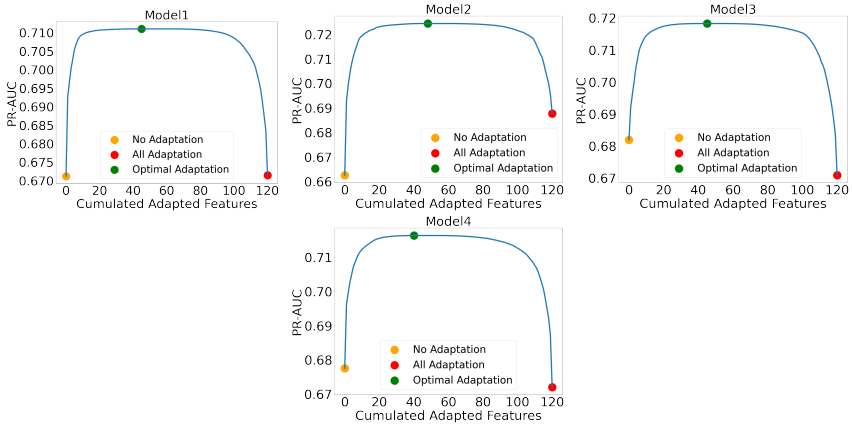

Fig. 5. Feature selection greedy algorithm with GBDT source model on Kaggle fraud detection task (idealized scenario involving all target labels).

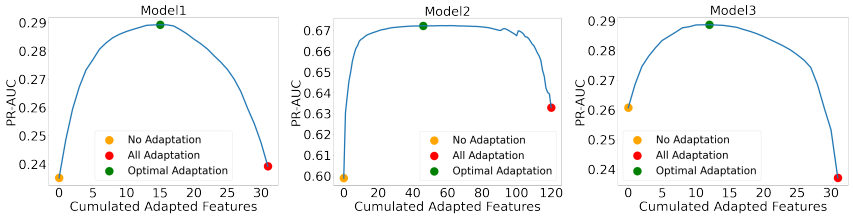

Fig. 6. Feature selection greedy algorithm with neural networks source model on Real Fraud dataset (idealized scenario involving all target labels.